Simple, Scalable Protocols for High-Performance Local Networks*

Rolf Riesen
Scalable Computing Systems
Sandia National Laboratories, P.O. Box 5800
Albuquerque, NM 87185-1110
rolf@cs.sandia.gov

Arthur B. Maccabe
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131-1386
maccabe@cs.unm.edu

RMPP (Reliable Message Passing Protocol) is a lightweight transport protocol designed for clusters. RMPP does try to avoid network congestion, but does provide end-to-end flow control and fault tolerance. We compare RMPP to TCP, UDP, and "Utopia," a simplistic protocol that provides no error recovery.

Figure 1 illustrates the framework used in this study. The benchmarks use a simple library that includes operations to pre-post a receive, send a message, and wait. Each protocol is configured to use a common *packet module*.

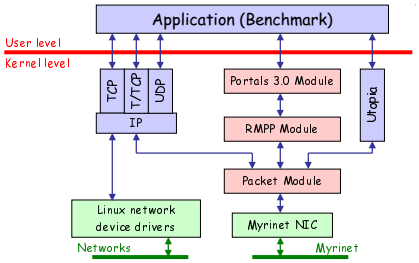

Figure 1. Protocol comparison framework

We start by comparing the protocols on four benchmarks: bandwidth, latency, all-to-all, and communication-computation overlap. We conclude by comparing TCP and RMPP for networks with very low bit error rates. The experiments were conducted on a 128 node Cplant [3] system at Sandia National Laboratories. Each compute node has 500 MHz Alpha EV6 (21264) CPU. The nodes are connected by a Myrinet network.

Figure 2 presents the bandwidth results. Utopia achieves the highest bandwidth. RMPP is second, attaining approximately 90% of Utopia's bandwidth. UDP initially achieves about 85% of Utopia's bandwidth and approaches RMPP for large messages. TCP's performance is somewhat erratic, varying between 70 and 77% of Utopia's performance.

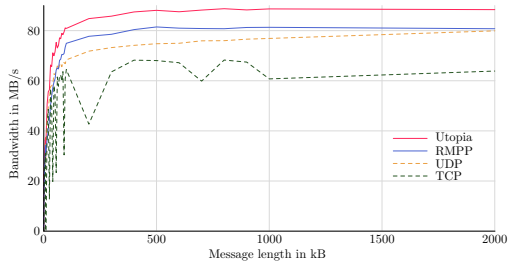

Figure 2. Bandwidth measurement

Table 1 summarizes the latency results. The RMPP latency measurements include the time needed to setup and tear down a connection for each message. In contrast, the connections for TCP are established during initialization and not included by our benchmark. Packet counts show that RMPP received three times as many packets as TCP.

Protocol	min	avg	max	std dev
Utopia	30 μs	33 μs	36 μs	0.913150
UDP	51 μs	52 μs	58 μs	1.221707
RMPP	65 μs	67 μs	72 μs	1.742920
TCP	61 μs	63 μs	65 μs	1.104917

Table 1. Latency measurements

Figure 3 presents the results of our all-to-all benchmark for 60KB messages. The lines starting in the upper left show the per-node bandwidth. TCP starts out higher, but drops as the number of nodes increases. The lines starting in the lower left show the number of packets sent by each protocol. The bars show the number of system errors that occurred during the application run. These errors are indicative of network congestion. (Myrinet switches drop mes-

* This work supported by Sandia, a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC004-94AL85000.

sages when the end bit of a message has not been seen for a certain amount of time.) The bar for RMPP is to the left and that for TCP is on the right. For 60KB messages, RMPP induces significantly less congestion. As the message size is increased, RMPP and TCP induce comparable congestion.

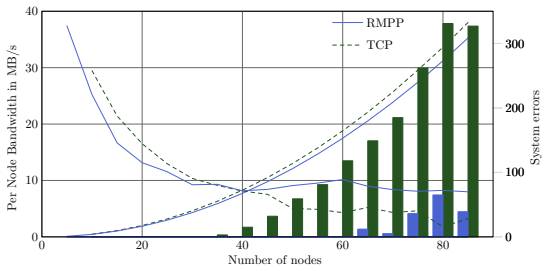

Figure 3. All-to-all, 60kB messages

The next benchmark measures the ability of a protocol to support communication during computation. This benchmark uses two processes: the first pre-posts a receive, performs a lengthy computation, and waits for a message; the second sends a message. The benchmark reports the waiting time the for the first process. Figure 4 presents the results for this benchmark. The waiting time for TCP and UDP are proportional to the message length. In contrast, the waiting times for RMPP and Utopia are nearly zero for all message sizes. These observations, combined with that fact that all protocols send same amount of data on the wire, lead us to conclude that Utopia and RMPP support overlapping computation and communication while TCP and UDP do not.

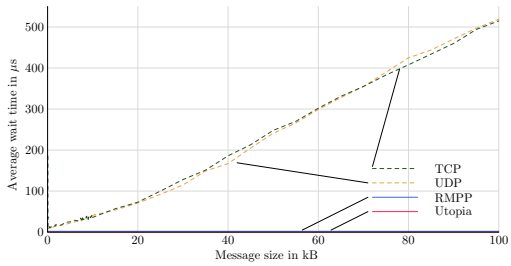

Figure 4. Overlap measurement

RMPP was designed for networks with very small bit error rates. Figure 5 presents a graphical interpretation of the goals for RMPP. RMPP should be better than TCP as long the bit error rate is very low. As the error rate increases, a dramatic decrease in performance is acceptable. In contrast, TCP performance would also suffer but should degrade more gracefully.

To explore this aspect of RMPP, we designed a simple client/server benchmark that measures client waiting time.

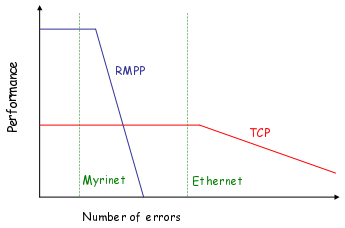

Figure 5. Expected behavior

Figure 6 shows the result for the client/server benchmark. The plot shows that after a threshold in the error rate is exceeded, the client wait time becomes worse.

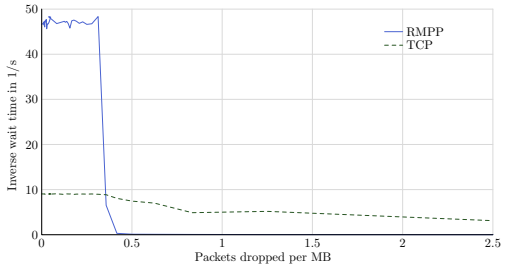

Figure 6. Observed behavior

Simple, message-based protocols like RMPP have several advantages over TCP, including: ease of implementation and testing, support for computation/communication overlap, and low CPU overhead. It was somewhat surprising to note that RMPP's lack of congestion control is not a hindrance. The results presented here and in [4] show that for large scientific clusters with high-performance networks, a simple message-based protocol like RMPP is a good choice.

References

[1] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W.-K. Su. Myrinet: A gigabit-per-second local-area-network. *IEEE Micro*, 15(1):29–36, Feb. 1995.

[2] R. Brightwell, T. Hudson, R. Riesen, and A. B. Maccabe. The Portals 3.0 message passing interface. Technical report SAND99-2959, Sandia National Laboratories, 1999.

[3] The Cplant Project Homepage, 2001. `http://www.cs.sandia.gov/cplant/`.

[4] R. Riesen. *Message-Based, Error-Correcting Protocols for Scalable High-Performance Networks*. PhD thesis, University of New Mexico, Albuquerque, New Mexico, July 2002.

[5] R. Riesen and A. B. Maccabe. RMPP: The reliable message passing protocol. In *Workshop on High-Speed Local Networks HSLN'02*, Tampa, Florida, Nov. 2002.